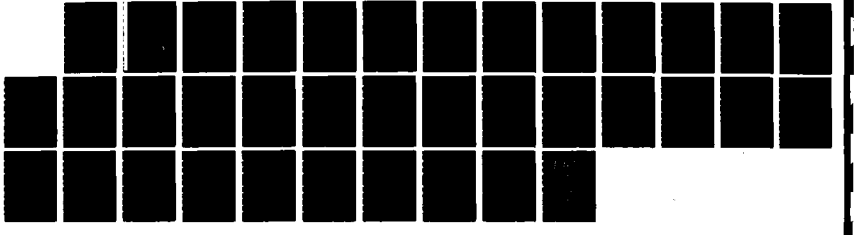
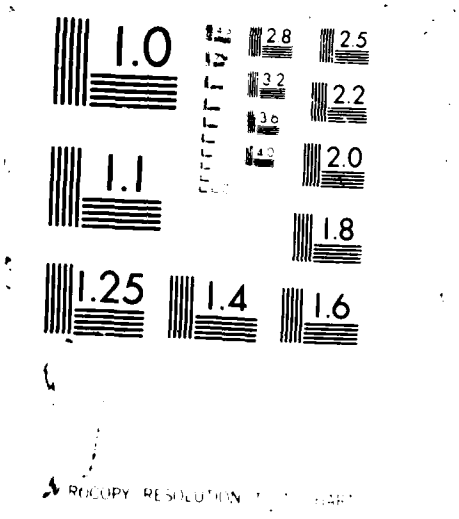
AD-A186 001 AN O(N3L) INTERIOR POINT ALGORITHM FOR CONVEX QUADRATIC 1/1
PROGRAMMING(U) CALIFORNIA UNIV BERKELEY OPERATIONS
RESEARCH CENTER R C MONTEIRO ET AL. JUN 87 ORC-87-15
UNCLASSIFIED N00014-87-C-0202 F/G 12/4 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
1.1
2.0
1.8
1.25
1.4
1.6

(12)

OPERATIONS RESEARCH CENTER

An $O(n^3L)$ Interior Point Algorithm
for Convex Quadratic Programming*

by

R. C. Monteiro**
I. Adler**

ORC 87-15 June 1987

UNIVERSITY OF CALIFORNIA

BERKELEY

(12)

An $O(n^3L)$ Interior Point Algorithm for Convex Quadratic Programming*

by

R. C. Monteiro**
I. Adler**

ORC 87-15 June 1987

*This research was partially funded by the Brazilian Post-Graduate Education Federal Agency (CAPES) and by the United States Navy Office of Naval Research under contract N00014-87-0202. Their financial support is gratefully acknowledged.

**Industrial Engineering and Operations Research Dept., University of California Berkeley, Berkeley CA 94720.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ORC 87-15	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER A186001
4. TITLE (and Subtitle) An $O(n^3L)$ Interior Point Algorithm for Convex Quadratic Programming		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER 87-15
7. AUTHOR(s) R. C. Monteiro I. Adler		8. CONTRACT OR GRANT NUMBER(s) N00014-87-0202
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operations Research Center 3115 Etcheverry Hall U.C. Berkeley Berkeley, CA 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Dept. of the Navy Arlingotn, VA 22217		12. REPORT DATE June 1987 13. NUMBER OF PAGES 32
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution Unlimited; Approved for public release

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(see report)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

An O(n³L) Interior Point Algorithm for Convex Quadratic Programming

R. C. Monteiro and I. Adler

Department of Industrial Engineering and Operations Research
University of California
Berkeley, CA 94720

August 1987

Abstract – We describe a primal-dual interior point algorithm for convex quadratic programming problems which requires a total of $O(n^3L)$ arithmetic operations. Each iteration updates a penalty parameter and finds an approximate Newton's direction associated with the Kuhn-Tucker system of equations which characterizes a solution of the logarithm barrier function problem. This direction is then used to find the next iterate. The algorithm is based on the path following idea. The total number of iterations is shown to be of the order of $O(\sqrt{n}L)$.

Key Words – Interior-point methods; Convex Quadratic Programming; Karmarkar's algorithm, Polynomial-time algorithms, Barrier function, Path following.

1. Introduction

In Monteiro and Adler [12], an algorithm to solve Linear Programming problems has been presented which converges in $O(\sqrt{n}L)$ iterations with an average number of $O(n^{2.5})$ arithmetic operations per iteration. In the last section of that paper, the authors observed that the same techniques could be extended to solve convex Quadratic Programming problems in at most $O(\sqrt{n}L)$ iterations. The objective of the present paper is to present the details of the algorithm mentioned in [12] as applied to convex Quadratic Programming problems.

Quadratic Programming (QP) problems share many of the combinatorial properties of Linear Programming (LP) problems. Based on these properties, algorithms extending the simplex method have been devised to solve QP problems. However, in the worst case, these algorithms may converge in an exponential number of steps.

Polynomial-time algorithms for convex Quadratic Programming problems based on the ellipsoid method were presented in [1] and [10].

Recently, with the advent of the new interior point algorithm by Karmarkar [7] for solving LP problems, some attention has been devoted to study classes of problems that can be solved by interior point algorithms in polynomial time. Ye and Tse [15] present an interior point algorithm for solving convex QP problems based on Karmarkar's projective transformation. Their algorithm is shown to converge in at most $O(nL)$ iterations with a computational effort of $O(n^3L^2)$ arithmetic operations per iteration. Thus overall their algorithm involves $O(n^4L^3)$ arithmetic operations in the worst case.

The algorithm discussed in this paper is based on the logarithm barrier function method and on the idea of following the path of minimizers for the logarithm barrier family problems, that is, the so called "central path". This path has been extensively studied in Bayer & Lagarias [2] and Meggido [11]. The logarithm barrier function approach is usually attributed to Frisch [4] and is formally studied in [3] in the context of nonlinear optimization. Algorithms for LP problems based on following the central path have been presented in [13], [14], [6] and [12]. The breakthrough in this line of research was obtained by Renegar [13], who was the first to achieve a speed of convergence of $O(\sqrt{n}L)$ iterations, where each iteration involves $O(n^3)$ arithmetic operations. His algorithm is based on the method of centers following the central path. Subsequently, Vaidya [14] improved Renegar's complexity to a total of $O(n^3L)$ arithmetic operations using the same approach of the method of centers. His algorithm converges with the same order of iterations as in Renegar's algorithm, however, he showed that the average number of arithmetic operations per iteration can be bounded by $O(n^{2.5})$. Independently, an equivalent complexity was also obtained by Gonzaga [6], using the logarithm barrier function approach. Both Vaidya's and Gonzaga's algorithms are primal algorithms. Kojima et al. [8] presented a primal-dual algorithm based on the logarithm barrier function method and the primal-dual framework described in [11]. Their algorithm is shown to converge in at most $O(nL)$ iterations with a computational

effort of $O(n^3)$ arithmetic operations per iteration, resulting in a total of $O(n^4L)$ arithmetic operations. Based on Kojima et al. [8] and on Gonzaga [6], a primal-dual algorithm converging in $O(\sqrt{n}L)$ iterations, in the worst case, with an average computational effort per iteration of $O(n^{2.5})$ arithmetic operations was presented by Monteiro and Adler [12]. As mentioned above, the current paper is an extension of these thechniques as applied to convex Quadratic Programming problems and it achieves a complexity similar to the Linear Programming case presented in [12].

Our paper is organized as follows. In section 2, we present some theoretical background. In section 3, we present the algorithm. In section 4, we prove results related to the convergence properties of the algorithm and we also describe the updating scheme that leads to a reduction in the average number of arithmetic operations per step. In section 5, we discuss how to initialize the algorithm. In section 6, we conclude the paper with some remarks.

2. Theoretical Background

In this section, we briefly review some theoretical results pertinent to the present work. A detailed discussion of these results can be found in [11]. We consider the convex quadratic programming problem as follows. Let

$$(P) \quad \min \; c^T x + \frac{1}{2} x^T Q x$$

$$s.t. \quad Ax = b$$

$$x \geq 0$$

where c, x are n-vectors, b is an m-vector, A is an $m \times n$ matrix and Q is a positive semi-definite $n \times n$ matrix. As for linear programming problems, the following fact is true for convex quadratic programming problems.

Proposition 2.1: *If problem (P) does not have an optimal solution then it must be either unbounded or infeasible.*

The Lagrangian dual problem corresponding to problem (P) is another quadratic programming problem given by

$$(D) \quad \max \quad -\frac{1}{2} v^T Q v + b^T y$$

$$s.t. \quad -Qv + A^T y + z = c$$

$$z \geq 0$$

where v and z are n-vectors and y is an m-vector. The relationship between problems (P) and (D) is provided by the following result known as the duality theorem for convex quadratic programming.

Proposition 2.2: *The following statements are true.*

(a) If problem (P) is unbounded then problem (D) is infeasible. If problem (D) is unbounded then problem (P) is infeasible.

(b) If problem (P) has an optimal solution x^o then there exist y^o and z^o such that the point $(v, y, z) = (x^o, y^o, z^o)$ is an optimal solution of problem (D). Conversely, if problem (D) has an optimal solution then problem (P) has an optimal solution. Moreover, the optimal values of both problems are identical.

The complementary slackness version for convex quadratic programming problems is as follows.

Proposition 2.3: *If x^o and (v^o, y^o, z^o) are optimal solutions for problems (P) and (D) respectively then*

$$(x^o)^T z^o = 0 \tag{2.1}$$

Conversely, if $(v, y, z) = (x^o, y^o, z^o)$ is a feasible solution of (D) such that x^o feasible for (P) and such that relation (2.1) holds, then x^o and (x^o, y^o, z^o) are optimal solutions of

problems (P) and (D) respectively.

The algorithm we consider in this paper has its motivation on the application of the logarithm barrier function technique to problem (P). The logarithm barrier function method consists of a consideration of the family of problems

$$(P_\mu) \quad \min \; c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^{n} \ln x_j$$

$$s.t. \; Ax = b$$

$$x > 0$$

where $\mu > 0$ is the barrier penalty parameter. This technique is well-known in the context of general constraint optimization problems. One solves the problem penalized by the logarithm barrier function term for several values of the parameter μ, with μ decreasing to zero, and the result is a sequence of feasible points converging to a "solution" of the original problem. This method is usually attributed to Frisch [4]. The interested reader can refer to Fiacco & McCormick [3] for a detailed discussion of this technique in the context of non-linear constrained optimization. Recently this method was first reconsidered in [5] where a similarity with Karmarkar's algorithm is discussed. Meggido [11] gives a comprehensive analysis of the logarithm barrier function approach as applied to Linear Programming and Linear Complementary problems with positive semi-definite matrices.

Before we can apply the logarithm barrier function method, some assumptions on the problems (P) and (D) are necessary. We impose the following assumptions:

Assumption 2.4:

(a) *The set* $S \equiv \left\{ x \in \mathbb{R}^n \; ; \; Ax = b, \; x > 0 \right\}$ *is non-empty.*

(b) *The set* $T \equiv \left\{ (v, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \; ; \; -Qv + A^T y + z = c \; , \; z > 0 \right\}$ *is non-empty.*

(c) *rank* $(A) = m$.

We say that points in the sets S and T are interior feasible solutions of problems (P) and (D) respectively. The need for (a) is evident since the logarithm barrier function method always works in the interior of the set defined by the inequality constraints. Assumptions (b) and (c) are also necessary as will become clear from the discussion that follows. In section 5, we will show how one can transform any given problem to one satisfying assumption 2.4.

Throughout this paper, we use the following notation. If $x = (x_1, \ldots, x_n)^T$ is an n-vector, then the corresponding capital letter X denotes the diagonal matrix $diag(x_1, \ldots, x_n)$. Observe that the objective function of problem (P_μ) is a strictly convex function. This implies that the problem (P_μ) has at most one global minimum, and that this global minimum, if it exists, is completely characterized by the Karush-Kuhn-Tucker stationary condition:

$$c + Qx - \mu X^{-1}e - A^T y = 0$$

$$Ax = b \quad , \quad x > 0$$

where e denotes the n-vector of ones and y is the Lagrangian multiplier associated with the equality constraints of problem (P_μ). By introducing the n-vector z, this system can be rewritten in an equivalent way as

$$(i) \quad ZXe - \mu e = 0 \tag{2.2}$$

$$(ii) \quad Ax = b \ , \ x > 0$$

$$(iii) \quad -Qx + A^T y + z = c$$

A necessary and sufficient condition for the problem (P_μ) to have a solution for all $\mu > 0$ is given by the following result.

Proposition 2.5 : *Assume (a) of Assumption 2.4 holds and let $\mu > 0$ be given. Then problem (P_μ) has an optimal solution if, and only if, the set of optimal solutions of problem (P) is non-empty and bounded.*

From this result, we immediately conclude that if (P_μ) has a solution for some $\mu > 0$ then it has a solution for all $\mu > 0$. The role played by assumption (b) is now provided by the following result.

Proposition 2.6 : *Assume that problem (P) is feasible. Then the set of optimal solutions of problem (P) is non-empty and bounded if, and only if, assumption (b) holds, that is, the set of interior feasible solutions of the dual problem (D) is non-empty.*

As a consequence of the previous two propositions, we have the the following corollary.

Corollary 2.7 : *Under assumptions (a) and (b), problem (P_μ) (and consequently the system (2.2)) has a unique solution $x(\mu)$, for all $\mu > 0$.*

The Kuhn-Tucker system (2.2) provides important information which we now point out. Assume that $\mu > 0$ is fixed in the system (2.2). Since $x > 0$, the first equation in the system (2.2) implies that $z > 0$. The third equation in (2.2) then implies that the triple (x, y, z) is an interior feasible solution for the dual problem (D). From assumption (c), it follows that there is a unique y satisfying (2.2). We denote the unique triple that satisfies (2.2) by $w(\mu) = (x(\mu), y(\mu), z(\mu))$. Consider the set W defined by

$$W = \left\{ (x, y, z) \in \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^n \; ; \; Ax = b \; , \; x > 0 \; , \; -Qx + A^T y + z = c \; , \; z > 0 \right\}$$

Observe that W is the set consisting of the interior feasible solutions (x, y, z) of problem (D) such that x is an interior feasible solution of problem (P). Obviously $w(\mu)$ is in the set W. The duality gap at point $w \in W$ is by definition given by

$$g(w) = c^T x + x^T Q x - b^T y$$

which is simply the value of the objective function of the problem (P) at x minus the value of the objective function of the problem (D) at (x, y, z). From the definition of the set W, one can easily verify that if $w \in W$ then

$$g(w) = x^T z \tag{2.3}$$

In particular, using the first equation in (2.2), it follows that

$$g(w(\mu)) = n\mu$$

for all μ and therefore $g(w(\mu))$ converges to zero as μ approaches zero. This implies that the objective function value of problem (P) at $x(\mu)$ and the objective function value of problem (D) at $(x(\mu), y(\mu), z(\mu))$ converge to the common optimal value of problems (P) and (D). In fact, the following stronger result holds true (c.f [11]).

Proposition 2.8 : *Under assumptions (a), (b) and (c), as* $\mu \to 0$, $x(\mu)$ *and* $(x(\mu), y(\mu), z(\mu))$ *converge to optimal solutions of problems (P) and (D) respectively.*

The following notation will be useful later. Let $w = (x, y, z) \in W$. We denote by $f(w) = (f_1(w), \ldots, f_n(w))^T \in \mathbb{R}^n$ the n-vector defined by

$$f_i(w) = x_i z_i \quad , \quad i = 1, \ldots, n$$

We denote by Γ the set (or path) of solutions $w(\mu)$, $\mu > 0$ for the system (2.2), i.e.,

$$\Gamma = \left\{ w(\mu) \equiv (x(\mu), y(\mu), z(\mu)) \; ; \; \mu > 0 \right\}.$$

The algorithm which will be presented in the next section is based on the idea of following this path Γ closely. The path Γ will serve as a criterion to guide the points generated by the algorithm.

3. The Algorithm

The algorithm presented in this section parallels the one presented in [12] for Linear Programming problems. We refer the reader to [8] and [12] for a motivation of the directions generated by the algorithm that we now describe. The directions generated by the algorithm are determined as follows. Given a point $w = (x, y, z)$ in the set $\mathcal{W}$, we consider the direction $\Delta w \equiv (\Delta x, \Delta y, \Delta z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ determined by the following system of linear equations

$$\tilde{Z}\Delta x + \tilde{X}\Delta z = XZe - \hat{\mu}e \tag{3.1.a}$$

$$A\Delta x = 0 \tag{3.1.b}$$

$$-Q\Delta x + A^T\Delta y + \Delta z = 0 \tag{3.1.c}$$

where $\tilde{x}$ and $\tilde{z}$ are vectors with all components positive and $\hat{\mu} > 0$ is some prespecified penalty parameter. *When $x = \tilde{x}$ and $z = \tilde{z}$, the direction Δw is exactly the Newton's direction* associated with the system (2.2). In the algorithm given below, we let $\tilde{x}$ and $\tilde{z}$ be approximations of the points x and z. The criterion of approximation is given in step 2 of algorithm 3.1. After some algebra, one obtains the following expressions for Δx and Δy.

$$\Delta x = (\tilde{Z} + \tilde{X}Q)^{-1}\left[I - \tilde{X}A^T\left(A(\tilde{Z} + \tilde{X}Q)^{-1}\tilde{X}A^T\right)^{-1}A(\tilde{Z} + \tilde{X}Q)^{-1}\right](XZe - \hat{\mu}e)$$

$$\Delta y = -\left[\left(A(\tilde{Z} + \tilde{X}Q)^{-1}\tilde{X}A^T\right)^{-1}A(\tilde{Z} + \tilde{X}Q)^{-1}\right](XZe - \hat{\mu}e)$$

Before describing the algorithm, some notations are necessary. Let $\tilde{s}$ denote the pair of approximations $(\tilde{x}, \tilde{z})$. We denote the direction $\Delta w = (\Delta x, \Delta y, \Delta z)$ determined by the system (3.1) as

$$\Delta w(w, \tilde{s}, \hat{\mu})$$

in order to indicate its dependence on the point $w = (x, y, z)$, on the approximation $\tilde{s}$ and on the penalty parameter $\hat{\mu}$

We are now ready to describe the algorithm. At the beginning of the algorithm, we assume that an initial point $w^o \equiv (x^o, y^o, z^o) \in W$ is available such that the following criterion of closeness with respect to the path Γ is satisfied:

$$\| f(w^o) - \mu^o e \| \leq \theta \mu^o \tag{3.2}$$

where $\| \cdot \|$ denotes the Euclidean norm, μ^o is a positive constant and $\theta = 0.1$.

We now state the algorithm.

Algorithm 3.1 :

Step 0) *Let $w^o \in W$ and $\mu^o > 0$ satisfy (3.2). Let ϵ be a given tolerance for the duality gap. Let*

$$\delta := 0.1$$

$$\gamma := 0.1 \tag{3.3}$$

Set k:=0.

Step 1) *If $g(w^k) = x^{kT} z^k < \epsilon$, stop.*

Step 2) *Choose $\tilde{s} = (\tilde{x}, \tilde{z})$ in $\mathbf{R}^n_+ \times \mathbf{R}^n_+$. satisfying:*

$$\frac{| x_i^k - \tilde{x}_i |}{| \tilde{x}_i |} \leq \gamma \ , \ i = 1, \ldots, n$$

$$\frac{| z_i^k - \tilde{z}_i |}{| \tilde{z}_i |} \leq \gamma \ , \ i = 1, \ldots, n$$

Step 3) *Set $\mu^{k+1} := \mu^k (1 - \delta / \sqrt{n})$.*

Calculate $\Delta w^k \equiv \Delta w(w^k, \tilde{s}, \mu^{k+1})$.

Step 4) *Set $w^{k+1} := w^k - \Delta w^k$.*

Set $k := k + 1$ and go to step 1.

In the following sections, we prove that all points generated by algorithm 3.1 are in the set W and that they remain close to the path Γ in a sense to be described latter. We also show that it terminates in at most $O(\sqrt{n}\ \max(\log\varepsilon^{-1}, \log n, \log\mu^o))$ iterations. Finally, we present a suitable choice for the approximation point $\tilde{s} = (\tilde{x}, \tilde{z})$ (see step 2 of the algorithm 3.1) that will enable us to show that algorithm 3.1 performs no more than $O(n^3 \max(\log\varepsilon^{-1}, \log n, \log\mu^o))$ arithmetic operations until its termination.

4. Convergence Results

In this section, we present convergence results for the algorithm described in section 3. Similar convergence results for Linear Programming problems are presented in [12]. Since the proofs of some results in this section are exactly the same as for the Linear Programming case, *the interested reader is referred to [12] for a detailed discussion. We have* omitted the proofs of those results which follow without modification from [12].

Let $w = (x, y, z) \in W$, $\tilde{s} = (\tilde{x}, \tilde{z}) \in \mathbb{R}^n \times \mathbb{R}^n$ and $\hat{\mu} > 0$. Let $\Delta w = (\Delta x, \Delta y, \Delta z)$ be the direction $\Delta w(w, \tilde{s}, \hat{\mu})$. Consider the point defined by $\hat{w} = w - \Delta w$. The next result provides expressions for the product of complementary variables $f_i(\hat{w})$, $i = 1, \ldots, n$.

Proposition 4.1: **Let** w, $\tilde{s}$ *and* $\hat{w}$ *be as above. Then the following expressions hold.*

$$f_i(\hat{w}) = \hat{\mu} + \Delta x_i \Delta z_i + (\tilde{x}_i - x_i)\Delta z_i + (\tilde{z}_i - z_i)\Delta x_i \tag{4.1}$$

$$(\Delta x)^T(\Delta z) \geq 0 \tag{4.2}$$

Proof: Expression (4.1) can be easily proved using the definition of $f_i(\hat{w})$ and expression (3.1.a). Multiplying expression (3.1.c) on the left by $(\Delta x)^T$, we obtain

$$(A\Delta x)^T\Delta y + (\Delta x)^T\Delta z - (\Delta x)^T Q\Delta x = 0 \tag{4.3}$$

Relations (3.1.b), (4.3) and the fact that the matrix Q is positive semidefinite immediately imply (4.2). This completes the proof of the proposition. □

We now state and prove a result that provides bounds necessary to show that the points generated by algorithm 3.1 are feasible and remain close to the path Γ.

Let $w = (x, y, z) \in \mathcal{W}$, $\tilde{s} = (\tilde{x}, \tilde{z}) \in \mathbb{R}^n \times \mathbb{R}^n$ and $\hat{\mu} > 0$. Let $\Delta w = (\Delta x, \Delta y, \Delta z)$ be the direction $\Delta w(w, \tilde{s}, \hat{\mu})$. We denote by $\Delta f \equiv (\Delta f_1, \ldots, \Delta f_n)^T$ the n-vector defined as

$$\Delta f = (\Delta x_1 \Delta z_1, \ldots, \Delta x_n \Delta z_n)^T \tag{4.4}$$

where Δx_i and Δz_i denotes the i^{th} coordinate of the vectors Δx and Δz respectively. The next result provides an upper bound on the Euclidean norm of the vector Δf.

Lemma 4.2: *Let Δf be defined as in (4.4). Then, we have*

$$\| \Delta f \| \leq \frac{\| f(w) - \hat{\mu} e \|^2}{2 \tilde{f}_{\min}} \tag{4.5}$$

where

$$\tilde{f}_{\min} \equiv \min \left\{ | \tilde{x}_i \tilde{z}_i | \; ; \; i = 1, \ldots, n \right\} \tag{4.6}$$

Furthermore, we have

$$\| \tilde{D} \Delta z \|^2 \leq \frac{\| f(w) - \hat{\mu} e \|^2}{\tilde{f}_{\min}} \tag{4.7}$$

$$\| \tilde{D}^{-1} \Delta x \|^2 \leq \frac{\| f(w) - \hat{\mu} e \|^2}{\tilde{f}_{\min}} \tag{4.8}$$

where $\tilde{D}$ is the diagonal matrix defined by

$$\tilde{D} = (\tilde{Z}^{-1} \tilde{X})^{1/2} \tag{4.9}$$

Proof: By equation (3.1.a), we have

$$\tilde{D}^{-1}\Delta x + \tilde{D}\Delta z = (\tilde{X}\tilde{Z})^{-1/2}(XZ - \hat{\mu}e) \tag{4.10}$$

From (4.2) it follows that

$$(\tilde{D}^{-1}\Delta x)^T(\tilde{D}\Delta z) \geq 0 \tag{4.11}$$

Using relations (4.10), (4.11) and the definition of the Euclidean norm, we obtain

$$\begin{aligned} \| \tilde{D}^{-1}\Delta x \|^2 + \| \tilde{D}\Delta z \|^2 &\leq \| \tilde{D}^{-1}\Delta x \|^2 + 2(\tilde{D}^{-1}\Delta x)^T(\tilde{D}\Delta z) + \| \tilde{D}\Delta z \|^2 \\ &= \|\tilde{D}^{-1}\Delta x + \tilde{D}\Delta z\|^2 \\ &= \| (\tilde{X}\tilde{Z})^{-1/2}(XZe - \hat{\mu}e) \|^2 \\ &= \sum_{i=1}^{n} \frac{(f_i(w) - \hat{\mu})^2}{|\tilde{x}_i\tilde{z}_i|} \\ &\leq \frac{\|f(w) - \hat{\mu}e\|^2}{\tilde{f}_{\min}} \end{aligned} \tag{4.12}$$

Inequalities (4.7) and (4.8) follow immediately from (4.12). Also (4.12) implies that

$$\| \tilde{D}^{-1}\Delta x \| \| \tilde{D}\Delta z \| \leq \frac{\|f(w) - \hat{\mu}e\|^2}{2\tilde{f}_{\min}} \tag{4.13}$$

On the other hand, using the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned} \| \Delta f \| &\leq \sum_{i=1}^{n} | \Delta x_i \Delta z_i | \\ &= \sum_{i=1}^{n} | \tilde{D}_{ii}^{-1}\Delta x_i | | \tilde{D}_{ii}\Delta z_i | \\ &\leq \| \tilde{D}^{-1}\Delta x \| \| \tilde{D}\Delta z \| \end{aligned} \tag{4.14}$$

Relations (4.13) and (4.14) imply inequality (4.5). This completes the proof of the lemma.

□

We now state the key result to prove the convergence of algorithm 3.1. We first introduce some notation. Given two vectors $x \in \mathbf{R}^n$ and $\bar{x} \in \mathbf{R}^n$, we denote the Euclidean norm of the vector $X^{-1}(\bar{x} - x)$ by $\|\bar{x} - x\|_x$, i.e.,

$$\|\bar{x} - x\|_x = \left[\sum_{i=1}^{n} \left(\frac{\bar{x}_i - x_i}{x_i} \right)^2 \right]^{1/2} \tag{4.15}$$

The key result is:

Theorem 4.3: **Let** $w = (x, y, z) \in W$ *and* $\mu > 0$ *satisfy*

$$\| f(w) - \mu e \| \leq \theta\mu \tag{4.16}$$

Let $\tilde{s} = (\tilde{x}, \tilde{z}) \in \mathbf{R}^n_+ \times \mathbf{R}^n_+$ *satisfy*

$$\frac{| x_i - \tilde{x}_i |}{| \tilde{x}_i |} \leq \gamma \ , \ i = 1, \ldots, n \tag{4.17}$$

$$\frac{| z_i - \tilde{z}_i |}{| \tilde{z}_i |} \leq \gamma \ , \ i = 1, \ldots, n \tag{4.18}$$

Let $\hat{\mu} > 0$ *be defined as*

$$\hat{\mu} = \mu(1 - \delta / \sqrt{n}) \tag{4.19}$$

Consider the point $\hat{w} \equiv (\hat{x}, \hat{y}, \hat{z}) \in \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^n$ *defined by*

$$\hat{w} \equiv w - \Delta w \tag{4.20}$$

where $\Delta w \equiv \Delta w(w, \tilde{s}, \hat{\mu})$. *Then the following hold:*

(a) The point $\hat{w}$ *is in the set* W *and satisfies*

$$\|\hat{x} - x\|_x \leq 0.28 \tag{4.21}$$

$$\|\hat{z} - z\|_z \leq 0.28 \tag{4.22}$$

(b) $\|f(\hat{w}) - \hat{\mu}e\| \leq \theta\hat{\mu}$

(c) $g(\hat{w}) \equiv \hat{x}^T\hat{z} \leq 1.1n\hat{\mu}$

The proof of theorem 4.3 is exactly the same as for the Linear Programming case and hence will not be given here (see lemma 4.8 and theorem 4.1 in section 4 of [12]). As a consequence of theorem 4.3, we have the following corollary.

Corollary 4.4: *All points w^k generated by algorithm 3.1 satisfy*

(a) w^k is in the set W, for all $k = 1,2,\ldots$ and

$$\| x^{k+1} - x^k \|_{x^k} \leq 0.28$$

$$\| z^{k+1} - z^k \|_{z^k} \leq 0.28$$

(b) $\| f(w^k) - \mu^k e \| \leq \theta\mu^k$, for all $k = 1,2,\ldots$

(c) $g(w^k) \equiv x^{kT}z^k \leq 1.1n\mu^k$, for all $k = 1,2,\ldots$

where

$$\mu^k = \mu^o(1 - \delta /\sqrt{n})^k \quad \text{for} \quad k = 1,2, \ldots$$

Proof: This result follows trivially by arguing inductively and using theorem 4.3. □

We now derive an upper bound on the total number of iterations performed by algorithm 3.1. The following result follows easily from Corollary 4.4 and is proved in section 4 of [12].

Proposition 4.5: *The total number of iterations performed by algorithm 3.1 is no greater than $k^* \equiv \lceil \log(1.1n\varepsilon^{-1}\mu^o)\sqrt{n} / \delta \rceil$ where $\varepsilon > 0$ denotes the tolerance for the duality gap and μ^o is the initial penalty parameter.*

With respect to the data of problem (P), define

$$L = \log_2 \left(\begin{array}{c} \textit{largest absolute value of the determinant} \\ \textit{of any square submatrix of } M \end{array} \right)$$

$$+ \log_2 (\max_j |c_j|) + \log_2 (\max_i |b_i|) + \log_2 (m + n) \tag{4.23}$$

where M is the matrix given by

$$M = \begin{bmatrix} Q & A \\ A^T & 0 \end{bmatrix} \tag{4.24}$$

It is straightforward to verify that the constant L is is less than two times the number of bits necessary to represent the data of problem (P). The following result says that we can find optimal solutions for problems (P) and (D) in $O(n^3)$ arithmetic operations once the duality gap at a point w^k generated by algorithm 3.1 becomes sufficiently small.

Proposition 4.6: *Let $w = (x, y, z)$ be a point in the set W satisfying*

$$x^T z \leq \frac{2^{-2L}}{(m + n)^2} \tag{4.25}$$

Then we can find a point $w^* = (x^*, y^*, z^*)$ in no more than $O(n^3)$ arithmetic operations, such that x^* and $w^* = (x^*, y^*, z^*)$ solve problems (P) and (D) respectively.

This result can be proved by slightly modifying the arguments of lemma 2 of [1]. Using this result, we obtain

Corollary 4.7: *If the initial penalty parameter μ^o satisfies $\log \mu^o = O(L)$ then algorithm 3.1 solves problem (P) in at most $O(\sqrt{n}L)$ iterations.*

Proof: Using the previous proposition, we can set $\epsilon = 2^{-2L}/(m + n)^2$ as the tolerance for the duality gap in algorithm 3.1. From proposition 4.5, we immediately conclude the validity of this corollary. □

In section 5, we will see that the initial penalty parameter μ^o can be chosen to satisfy $\log \mu^o = O(L)$. One possible choice for the approximation $\tilde{s} \equiv (\tilde{x}, \tilde{z})$ on step 2 of the algorithm 3.1 is to use exact data, that is, to set $\tilde{s}$, on the k^{th} iteration, equal to s^k. With

this choice of $\bar{s}$, we have the following result:

Corollary 4.8: *Algorithm 3.1 solves problem (P) in no more than* $O(n^{3.5}L)$ *iterations.*

Proof: At every iteration, the computational effort is majorized by the calculation of the inverse of the matrix $[A(Z^k + X^k Q)^{-1} X^k A^T]$ which requires $O(n^3)$ arithmetic operations. By corollary 4.7, algorithm 3.1 terminates in at most $O(\sqrt{n}L)$ iterations. These two observations immediately concludes the proof of the corollary. □

In the next subsection, we present an alternative choice for the approximation that makes possible to reduce the complexity of algorithm 3.1 to $O(n^3L)$ arithmetic operations.

4.1. A Good Choice for $\bar{x}$ and $\bar{z}$

In this subsection, we show that the complexity of algorithm 3.1 can be reduced to $O(n^3L)$ arithmetic operations. The arguments in this subsection are the same as for the linear programming case presented in section 5 of [12]. We should point out that this idea for reduction of the complexity was first presented in Karmarkar [7] and subsequently in Gonzaga [6] and Vaidya [14]. The reduction basically consists of using a direction that approximates the "exact" direction calculated from using "exact" data, that is, the current iterate. In our case, an approximate direction is implicit in the choice of the approximation $\bar{s}$. In this section, we show that by choosing the approximation $\bar{s}$ conveniently, a reduction in the average work per iteration is obtained. The choice of the approximation $\bar{s}$ is made by an updating scheme as follows (In the procedure below, k stands for the iteration count):

Updating scheme 4.9:

For $k := 0$, *set* $\bar{x} := x^0$ *and* $\bar{z} := z^0$

For $k > 0$ *do*

For $i = 1, \ldots, n$ *do*

If one of the following holds

$$\text{(a)} \quad \frac{| x_i^k - \tilde{x}_i |}{| \tilde{x}_i |} > \gamma$$

$$\text{(b)} \quad \frac{| z_i^k - \tilde{z}_i |}{| \tilde{z}_i |} > \gamma$$

then set $\tilde{x}_i := x_i^k$ *and* $\tilde{z}_i := z_i^k$.

Some observations are in order at this point. In order to calculate the directions Δx and Δy determined by system (3.1), we need to calculate the inverse of the matrix

$$A(\tilde{Z} + \tilde{X}Q)^{-1}\tilde{X}A^T = A(\tilde{X}^{-1}\tilde{Z} + Q)^{-1}A^T \tag{4.26}$$

where $\tilde{s} \equiv (\tilde{x}, \tilde{z})$ represents the approximation for the current iteration. Let $\tilde{s}^k$ and B_k denote the approximation $\tilde{s}$ and the matrix given by (4.26) respectively at the k^{th} iteration of the algorithm 3.1. Also let D_k denote the matrix $(\tilde{Z}^k)^{-1}\tilde{X}^k$. We show next that if the matrix D_k differs from the matrix D_{k-1} by exactly l diagonal elements then the computation of B_k^{-1} can be carried out in $O(n^2 l)$ arithmetic operations by means of l rank-one updates. Let $E = Q + D_{k-1}$ and $F = D_k - D_{k-1}$. Then we obtain

$$B_{k-1} = AE^{-1}A^T \tag{4.27}$$

and

$$B_k = A(E + F)^{-1}A^T \tag{4.28}$$

Obviously, F is a diagonal matrix. Denote the i^{th} diagonal element of the matrix F by f_i. By assumption, exactly l diagonal elements f_i are non-zero. For simplicity of notation, we assume that that these elements are the first l diagonal entries of the matrix F. Then the matrix F can be written as

$$F = \sum_{i=1}^{l} f_i u^i (u^i)^T$$

where u^i denotes the n-vector where all components are zero except the i^{th} component which equals one. Let E_j be defined as

$$E_o = E$$

$$E_j = E_{j-1} + f_j u^j (u^j)^T \quad , \quad j = 1, \ldots, l \tag{4.29}$$

Note that $E_l = E + F$. We observe that the matrices $E_o, E_1, \ldots, E_l$ are positive definite, and hence invertible matrices. Applying the well known Shermann-Morrison formula of Linear Algebra to the matrix E_j as given in expression (4.29), we obtain for $j = 1, \ldots, l$

$$E_j^{-1} = E_{j-1}^{-1} - \left(\frac{f_j}{1 + f_j (u^j)^T E_{j-1}^{-1} u^j} \right) E_{j-1}^{-1} u^j (u^j)^T E_{j-1}^{-1}$$

Recursively, we can obtain E_l^{-1} as follows.

$$E_l^{-1} = E_o^{-1} - \sum_{i=1}^{l} g_i v^i (v^i)^T \tag{4.30}$$

where the scalars g_i and the n-vectors v^i , $i = 1, \ldots, l$ are generated by the following iterative procedure.

Procedure 4.11: *Given E_o^{-1} then,*

For i = 1 ,...., l do

$$g_j = \frac{f_j}{1 + f_j (u^j)^T E_{j-1}^{-1} u^j}$$

$$v^j = E_{j-1}^{-1} u^j$$

$$E_j^{-1} = E_{j-1}^{-1} - g_j v^j (v^j)^T$$

Since $E_l = E + F$ and using expressions (4.27), (4.28) and (4.30), we obtain

$$B_k = B_{k-1} - \sum_{j=1}^{l} g_j (Av^j)(Av^j)^T \tag{4.31}$$

We can also use the same process described above to find the inverse of the matrix B_k using expression (4.31) and the matrix B_{k-1}^{-1} already calculated in the prev ıs iteration of the algorithm. We note that the procedure above involves $O(n^2 l)$ arithmetic operations.

Next we provide an upper bound on the number of diagonal element changes that occurs on the matrix $\tilde{Z}^{-1}\tilde{X}$ during K steps of algorithm 3.1. Note that the i^{th} diagonal element of the matrix $(\tilde{Z}^{-1}\tilde{X})$ changes only when inequality (a) or (b) of the updating scheme 4.9 is satisfied.

The following result can be proved by using the arguments in section 5 of [6].

Proposition 4.12: *Let $(v^k)_{k=0}^K$ be a sequence of n-vectors with all components positive and satisfying*

$$\|v^{k+1} - v^k\|_v \leq \rho \quad , \quad k = 0, 1, \ldots, K-1$$

where ρ is a positive constant less than one. Define the sequence $(\tilde{v}^k)_{k=0}^K$ recursively as follows. Set $\tilde{v}^o := v^o$ and for $k \geq 1$ and $i = 1, \ldots, n$ let

$$\tilde{v}_i^k := \begin{cases} v_i^k & \text{if} \quad \dfrac{|v_i^k - \tilde{v}_i^{k-1}|}{|\tilde{v}_i^{k-1}|} > \gamma \\ \tilde{v}_i^{k-1} & \text{otherwise} \end{cases}$$

where γ is a positive constant less than one. Let V_i^K be the set of indices k defined as

$$V_i^K = \left\{ k \; ; \; \frac{|v_i^k - \tilde{v}_i^{k-1}|}{|\tilde{v}_i^{k-1}|} > \gamma \, , \, 1 \leq k \leq K \right\}$$

and let $|V_i^K|$ denote its cardinality, that is, the number of times the i^{th} component of the sequence $(\tilde{v}^k)_{k=0}^K$ changes. Then the following inequality holds

$$\sum_{i=1}^{n} |V_i^K| \leq - \frac{\rho K \sqrt{n}}{(1-\rho)\log(1-\gamma)}$$

As a consequence of this result, we have the following corollary.

Corollary 4.13: *Let* $(x^k)_{k=0}^K$ *and* $(z^k)_{k=0}^K$ *be the sequences generated by algorithm 3.1 and consider the approximation* $\tilde{s} = (\tilde{x}, \tilde{z})$ *defined as in the updating scheme 4.9. Consider the following two sets:*

$$S_i^K = \left\{ k \; ; \; \frac{|x_i^k - \tilde{x}_i|}{|\tilde{x}_i|} > \gamma, \; 1 \leq k \leq K \right\}$$

$$T_i^K = \left\{ k \; ; \; \frac{|z_i^k - \tilde{z}_i|}{|\tilde{z}_i|} > \gamma, \; 1 \leq k \leq K \right\}$$

Then the following inequalities hold.

$$\sum_{i=1}^{n} |S_i^K| \leq 4.5\sqrt{n}K$$

$$\sum_{i=1}^{n} |T_i^K| \leq 4.5\sqrt{n}K$$

Proof: This result follows immediately by using relations (4.21), (4.22) and proposition 4.12. □

Thus, the total number of rank-one updates that occurs during K steps of algorithm 3.1 is on the order of $O(\sqrt{n}K)$. As a consequence of this result, we have :

Corollary 4.14: *Algorithm 3.1 coupled with the updating scheme 4.9 solves problem (P) in no more than* $O(n^3L)$ *arithmetic operations.*

Proof: From corollary 4.7, we know that algorithm 3.1 finds an optimal solution of problem (P) in $O(\sqrt{n}L)$ iterations. Corollary 4.13 implies that the total number of rank-one updates is then of the order of $O(nL)$. Since each rank-one update involves $O(n^2)$ arithmetic operations, the total number of arithmetic operations is then of the order of $O(n^3L)$. This completes the proof of the corollary. □

5. Initialization of the Algorithm

In this section, we show how to initialize algorithm 3.1, in order to solve any convex Quadratic Programming problem. The approach is to use a transformed problem equivalent to original one that satisfies the initial condition (3.3). Therefore, by solving the transformed problem, we are able to obtain a solution for the original problem. We also show that the "size" of the transformed problem is of the same order of the "size" of the original problem, where by "size", we mean the constant L defined in relation (4.23). This fact implies that, with respect to the data of the original problem, the complexity obtained in section 4 is still valid.

Consider the convex quadratic programming problem

$$(\tilde{P}) \quad \min \ \tilde{c}^T\tilde{x} + \frac{1}{2}\tilde{x}^T\tilde{Q}\,\tilde{x}$$

$$s.t. \ \tilde{A}\,\tilde{x} = \tilde{b}$$

$$\tilde{x} \geq 0$$

where $\tilde{A}$ is an $\tilde{m}\times\tilde{n}$ matrix which has full row rank, $\tilde{Q}$ is an $\tilde{n}\times\tilde{n}$ positive semi-definite matrix and $\tilde{b}$, $\tilde{c}$ are vectors of length $\tilde{m}$ and $\tilde{n}$ respectively. We assume that the entries of the vectors $\tilde{b}$ and $\tilde{c}$ and the entries of the matrices $\tilde{A}$ and $\tilde{Q}$ are integral. Let

$$\tilde{L} = \log_2 \left(\begin{array}{c} \textit{largest absolute value of the determinant} \\ \textit{of any square submatrix of } \tilde{M} \end{array} \right)$$

$$+ \log_2 (\max_j |\tilde{c}_j|) + \log_2 (\max_i |\tilde{b}_i|) + \log_2 (\tilde{m} + \tilde{n})$$

where $\tilde{M}$ is the $(\tilde{m}+\tilde{n}) \times (\tilde{m}+\tilde{n})$ matrix defined as

$$\tilde{M} = \begin{bmatrix} \tilde{Q} & \tilde{A}^T \\ \tilde{A} & 0 \end{bmatrix}$$

The Karush-Kuhn-Tucker necessary and sufficient condition for $\tilde{x} \in \mathbf{R}^{\tilde{n}}$ to be a solution of $(\tilde{P})$ is that there exists a vector $\tilde{y} \in \mathbf{R}^{\tilde{m}}$ such that $\tilde{x}$ and $\tilde{y}$ satisfy

$$\begin{aligned} &\tilde{z} \equiv \tilde{c} + \tilde{Q}\,\tilde{x} - \tilde{A}^T \tilde{y} \geq 0 \\ &\tilde{A}\,\tilde{x} = \tilde{b} \\ &\tilde{x} \geq 0 \\ &\tilde{z}^T \tilde{x} = 0 \end{aligned} \tag{5.1}$$

which can be rewritten as

$$\begin{pmatrix} \tilde{z} \\ 0 \end{pmatrix} - \begin{bmatrix} -\tilde{Q} & \tilde{A}^T \\ -\tilde{A} & 0 \end{bmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} \tilde{c} \\ -\tilde{b} \end{pmatrix} \tag{5.2.a}$$

$$\tilde{x} \geq 0 \quad , \quad \tilde{z} \geq 0 \tag{5.2.b}$$

$$\tilde{x}^T \tilde{z} = 0 \tag{5.2.c}$$

A well known result from Linear Complementary theory is that if the system (5.2) has some solution then it has a solution which is a vertex of the polyhedron given by (5.2.a) and (5.2.b). The following lemma is a well known result whose proof follows from an immediate application of Cramer's rule.

Lemma 5.1: *Let $\tilde{w} = (\tilde{x}, \tilde{y}, \tilde{z})$ be a vertex of the polyhedron given by (5.2.a) and (5.2.b). Then the coordinates of $\tilde{w}$ are rational numbers with numerator and denominator less than*

or equal to $2^{\tilde{L}}$.

We observe that any solution $(\tilde{x}, \tilde{y})$ of system (5.1) is an optimal solution of the dual of problem $(\tilde{P})$, which is the problem given by

$$(\tilde{D}) \quad \max \; -\frac{1}{2}\tilde{v}^T\tilde{Q}\,\tilde{v} + \tilde{b}^T\tilde{y}$$

$$-\tilde{Q}\,\tilde{v} + \tilde{A}^T\tilde{y} \le \tilde{c}$$

where $\tilde{v}$ is an $\tilde{n}$-vector and $\tilde{y}$ is an $\tilde{m}$-vector. Let $n = \tilde{n} + 2$ and $m = \tilde{m} + 1$. Let $\lambda = 2^{\tilde{L}}$ and $K > 0$ be a large constant which will be specified more precisely later. Consider the transformed problem as follows.

$$(P) \quad \min \; \tilde{c}^T\tilde{x} + \frac{1}{2}\tilde{x}^T\tilde{Q}\,\tilde{x} + K\,\tilde{x}_n$$

$$s.t. \quad \tilde{A}\,\tilde{x} + (\tilde{b} - \lambda\tilde{A}e)\tilde{x}_n = \tilde{b}$$

$$e^T\tilde{x} + \tilde{x}_{n-1} + \lambda\tilde{x}_n = n\lambda$$

$$\tilde{x} \ge 0 \;,\; \tilde{x}_{n-1} \ge 0 \;,\; \tilde{x}_n \ge 0$$

where $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_{n-2})^T$ is an $(n-2)$-vector and $\tilde{x}_{n-1}$ and $\tilde{x}_n$ are scalars. The dual problem corresponding to problem (P) is given by

$$(D) \quad \max \; -\frac{1}{2}\tilde{v}^T\tilde{Q}\,\tilde{v} + \tilde{b}^T\tilde{y} + (n\lambda)\tilde{y}_m$$

$$-\tilde{Q}\,\tilde{v} + \tilde{A}^T\tilde{y} + e\,\tilde{y}_m \le \tilde{c}$$

$$\tilde{y}_m \le 0$$

$$(\tilde{b} - \lambda\tilde{A}e)^T\tilde{y} + \tilde{y}_m \le K$$

where $\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_{m-1})^T$ is an $(m-1)$-vector, $\tilde{y}_m$ is a scalar and $\tilde{v}$ is an $(n-2)$-vector. These problems can be recast in the notation of problems (P) and (D) of section 2 as follows. Let $x = (\tilde{x}^T, \tilde{x}_{n-1}, \tilde{x}_n)^T \in \mathbb{R}^n$, $y = (\tilde{y}^T, \tilde{y}_m)^T \in \mathbb{R}^m$ and $v = (\tilde{v}^T, \tilde{v}_{n-1}, \tilde{v}_n)^T \in \mathbb{R}^n$

Define $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$ as follows.

$$b = \begin{pmatrix} \tilde{b} \\ n\lambda \end{pmatrix} \qquad c = \begin{pmatrix} \tilde{c} \\ 0 \\ K \end{pmatrix} \qquad A = \begin{bmatrix} \tilde{A} & 0 & \tilde{b} - \lambda \tilde{A} e \\ e^T & 1 & \lambda \end{bmatrix} \tag{5.3}$$

Let $Q \in \mathbb{R}^{n \times n}$ denote the block diagonal matrix as follows.

$$Q = diag(\tilde{Q}, 0, 0) \tag{5.4}$$

With these notations, we can then rewrite problems (P) and (D) as in section 2. We refer to these two formats interchangeably.

In the following, we adopt the convention to denote the optimal value of a quadratic programming problem (P) as $val(P)$ and the value of the objective function of (P) at a feasible point x as $val_P(x)$. We now present the relationship between the optimal solutions of the transformed problem (P), and its dual (D), with the optimal solutions of the original problem $(\tilde{P})$, and its dual $(\tilde{D})$, respectively. Before stating the relation, we make the following observation. If problem $(\tilde{P})$ has an optimal solution then, by the observations preceding lemma 5.1, a pair $(\tilde{x}_*, \tilde{y}_*)$ must exist which solves the system (5.2) and which is a vertex of the polyhedron defined by relations (5.2.a) and (5.2.b). The pair $(\tilde{x}_*, \tilde{y}_*)$ is considered in the following result.

Lemma 5.2: *Assume that problem $(\tilde{P})$ has an optimal solution and let $(\tilde{x}_*, \tilde{y}_*)$ be as above. Assume that the cost coefficient K in the cost vector of problem (P) satisfies*

$$K > (\tilde{b} - \lambda \tilde{A} e)^T \tilde{y}_* \tag{5.5}$$

Then the following statements hold.

(1) The common optimal value of problems (P) and (D) is equal to the common optimal value of problems $(\tilde{P})$ and $(\tilde{D})$.

(2) If $x = (x_1, \ldots, x_n)^T$ and $(v, y) = ((v_1, \ldots, v_n)^T, (y_1, \ldots, y_m)^T)$ are optimal solutions of problems (P) and (D) respectively, then $x_n = 0$ and $y_m = 0$. Moreover,

$\tilde{x} = (x_1, \ldots, x_{n-2})^T$ *and* $(\tilde{v}, \tilde{y}) = ((v_1, \ldots, v_{n-2})^T, (y_1, \ldots, y_{m-1})^T)$ *are optimal solutions of* $(\tilde{P})$ *and* $(\tilde{D})$.

Proof: We first prove (1). Consider the vectors $x_* \in \mathbf{R}^n$ and $y_* \in \mathbf{R}^m$ defined as follows.

$$x_* = (\tilde{x}_*^T, n\lambda - e^T\tilde{x}_*^T, 0)^T$$

$$y_* = (\tilde{y}_*^T, 0)^T$$

Expression (5.5) and lemma 5.1 imply that x_* and (x_*, y_*) are feasible solutions of problems (P) and (D) respectively. Thus, we have

$$val(\tilde{P}) = val_{\tilde{P}}(\tilde{x}_*) = val_P(x_*) \geq val(P) \tag{5.6}$$

and

$$val(\tilde{D}) = val_{\tilde{D}}(\tilde{x}_*, \tilde{y}_*) = val_D(x_*, y_*) \leq val(D) \tag{5.7}$$

Since $val(D) = val(P)$ and $val(\tilde{D}) = val(\tilde{P})$, relations (5.6) and (5.7) then immediately imply (1). Moreover, x_* and (x_*, y_*) are optimal solutions for (P) and (D) respectively.

We now prove (2). Since x_* and (v, y) form a pair of primal and dual optimal solutions for problems (P) and (D) respectively, they must satisfy the complementary slackness condition (c.f. proposition 2.2 of section 2). In particular, we have

$$(-y_m)\,(n\lambda - e^T\tilde{x}_*) = 0 \tag{5.8}$$

But lemma 5.1 implies that $e^T\tilde{x}_* \leq \tilde{n}\lambda = (n-2)\lambda < n\lambda$. Therefore, (5.8) implies that $y_m = 0$. But this implies that $(\tilde{v}, \tilde{y})$ is feasible to $(\tilde{D})$ and that $val_D(v, y) = val_{\tilde{D}}(\tilde{v}, \tilde{y})$. Statement (1) above then implies that $(\tilde{v}, \tilde{y})$ is optimal for $(\tilde{D})$.

Arguing with the pair x and (x_*, y_*) in a similar way, we conclude that $x_n = 0$ and that $\tilde{x}$ is optimal for $(\tilde{P})$. This proves (2). □

Lemma 5.3: *Assume that problem* $(\tilde{P})$ *has an optimal solution and let* $(\tilde{x}_*, \tilde{y}_*)$ *be as in lemma 5.2. Then* $K = 2^{3\tilde{L}}$ *satisfies relation (5.5).*

Proof: This lemma follows straightforwardly from the definition of $\tilde{L}$ and from lemma 5.1. □

In view of lemma 5.3, from now on, we let $K = 2^{3\tilde{L}}$. Consider the constant L defined as in relation (4.23) and (4.24). The following observations are easily shown.

(1) From the definition of Q, A, b and c given by expressions (5.3) and (5.4), it immediately follows that

$$\tilde{L} \leq L$$

(2) The largest absolute value of the determinant of any square submatrix of A is at most $(\tilde{m} + \tilde{n})2^{3\tilde{L}}$.

(3) $\max_{i=1,\ldots,m} |b_i| \leq n2^{\tilde{L}}$ and $\max_{j=1,\ldots,n} |c_j| \leq 2^{2\tilde{L}}$.

(4) Statements (2) and (3) implies that $L \leq 9\tilde{L}$.

We now verify that problem (P) satisfies assumption 2.1 of section 2. Assumption (c) is obviously satisfied since $\tilde{A}$ was assumed to have full row rank. We verify assumptions (a) and (b) jointly by exhibiting a point $w^o = (x^o, y^o, z^o)$ which is in the set W defined in section 2 and satisfying the criterion of closeness (3.3). Let $x^o \equiv (\lambda, \ldots, \lambda, 1)^T \in \mathbb{R}^n$. Let Q_j, $j = 1,\ldots,n$ denote the j^{th} row of the matrix Q and let $y^o = (0, \ldots, 0, -\mu^o/\lambda)^T \in \mathbb{R}^m$ where μ^o satisfies

$$\mu^o \geq \frac{\left[\sum_{j=1}^{n-1}(\lambda c_j + \lambda Q_j{}^T x^o)^2 + (c_n + Q_n{}^T x^o)^2\right]^{1/2}}{\theta} \tag{5.9}$$

Let $z^o \in \mathbb{R}^n$ denote the slack vector $c + Qx^o - A^T y^o$ for the dual (D) corresponding to the pair $v = x^o$ and $y = y^o$. Since $A^T y^o = \mu^o/\lambda\ (1, \ldots, 1, \lambda)^T \in \mathbb{R}^n$, it is easy to verify that

$$\sum_{j=1}^{n}(x_j^o z_j^o - \mu^o)^2 = \sum_{j=1}^{n-1}(\lambda c_j + \lambda Q_j^T x^o)^2 + (c_n + Q_n^T x^o)^2$$

and hence that the criterion of closeness (3.3) is satisfied due to expression (5.9). Since $\log\lambda = \tilde{L} \leq L$, it is straightforward to verify that the logarithm of right hand side of (5.9) is on the order of $O(L)$. Therefore μ^o can be chosen to satisfy $\log \mu^o = O(L)$. By the convergence results of section 4 and 5, it follows that problems (P) and (D) can be solved in at most $O(n^3L)$ arithmetic operations. Finally, the main result of this section is as follows.

Proposition 5.4: *Problem $(\tilde{P})$ can be solved in at most $O(\tilde{n}^3\tilde{L})$ arithmetic operations.*

Proof: Applying algorithm 3.1 to problem (P), we obtain vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ such that x and (x, y) are optimal solutions for problems (P) and (D) respectively. Consider the following two possible cases.

(i) If $x_n = 0$ and $y_m = 0$ then problem (P), and consequently (D), has an optimal solution. Indeed, if we let $\tilde{x} = (x_1, \ldots, x_{n-2})^T$ and $\tilde{y} = (y_1, \ldots, y_{m-1})^T$ then $\tilde{x}$ and $(\tilde{x}, \tilde{y})$ are feasible solutions for $(\tilde{P})$ and $(\tilde{D})$ respectively. Statement (2) of lemma 5.2 then implies that $\tilde{x}$ and $(\tilde{x}, \tilde{y})$ are optimal solutions for $(\tilde{P})$ and $(\tilde{D})$ respectively.

(ii) If either $x_n \neq 0$ or $y_m \neq 0$ then lemma 5.2 implies that $(\tilde{P})$ is either unbounded or infeasible. In this case we solve the LP problem obtained by replacing the objective function of problem (P) by the linear function Kx_n. If the resulting optimal solution of this problem satisfies $x_n = 0$ then $(\tilde{P})$ is unbounded. Otherwise, $(\tilde{P})$ is infeasible.

By corollary 4.14, the computation above can be carried out in at most $O(n^3L)$ arithmetic operations. Since $\tilde{n} = n - 2$ and $L \leq 9\tilde{L}$, the total number of arithmetic operations is on the order of $O(\tilde{n}^3\tilde{L})$. □

6. Remarks

The following observations are in order:

(1) The purpose of this paper is to present a theoretical result. Thus in order to simplify the presentation, we constructed $\hat{\mu} = \mu(1 - \delta/\sqrt{n})$. Obviously, one can use $\hat{\mu}$ which is less than or equal than the above one, but still satisfying (b) of theorem 4.3 and relations (4.21) and (4.22). In this way, one can accelerate the convergence of the algorithm.

(2) Additional improvements in actual implementation, which are possible, such as more judicious selection of θ, δ and γ, together with actual test results, are the subject of a forthcoming paper.

(3) With the necessary modifications, the results of this paper are also valid in the case that the matrix Q is only assumed to be positive semi-definite on the affine space $\{ x \mid Ax = b \}$. All the duality results remain true if we add the constraint $Ax = b$ to the dual problem (D). The formulas for the directions Δx and Δy given in section 3 do not necessarily hold in this case. We leave it to the reader to carry out the necessary modifications. Finally, we note that the convergence results of section 4 follow without any modifications.

(4) It is well known that a Linear Complementarity problem with positive semi-definite matrix can be reduced to an equivalent convex Quadratic Programming problem and vice-versa (c.f. [11]). Thus, the algorithm presented in this paper can be used to solve Linear Complementarity problems with positive semi-definite matrices. At the time of writing this paper, we were informed of a recent paper by Kojima et. al. [10] which present an algorithm for solving Linear Complementarity problems with positive semi-definite matrices. They obtained the same complexity as the one achieved in this paper.

7. Acknowledgement

This research was partially funded by the Brazilian Post-Graduate Education Federal Agency (CAPES) and by the United States Navy Office of Naval Research under contract N00014-87-K-0202. Their financial support is gratefully acknowledged.

References

[1] I. Adler, R. P. McLean, J. S. Provan, " An Application of the Khachian-Shor Algorithm to a Class of Linear Complementary Problems", *manuscript, 1980*

[2] K. Bayer, J. C. Lagarias, "The Nonlinear Geometry of Linear Programming", *preprints, AT&T Bell Laboratories*, Murray Hill, NJ, 1986

[3] A. Fiacco, G. McCormick, "Nonlinear Programming: Sequential Unconstrained Minimization Techniques, *John Wiley and Sons*, New York, 1955

[4] K.R. Frisch, "The Logarithm Potential Method of Convex Programming", *unpublished manuscript, University Institute of Economics*, Oslo, Norway,1955

[5] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin, M.H. Wright, "On Projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method", *Technical Report Sol 85-11, Systems Optimization Laboratory, Department of Operations Research, Stanford University*, Stanford, 1985

[6] C.C. Gonzaga, "An Algorithm for Solving Linear Programming Problems in $O(n^3L)$ Operations", *Memorandum Number UCB/ERL M87/10, Eletronics Research Laboratory, University of California*, Berkeley, CA March, 1987

[7] N. Karmarkar, "A New Polynomial Time Algorithm for Linear Programming", *Combinatorica*, vol 4, 373-395, 1984

[8] M. Kojima, S. Mizuno, A. Yoshise, "A Primal-Dual Interior Point Algorithm for Linear Programming", *Report No. B-188, Department of Information Sciences, Tokyo Institute of Technology*, Tokyo, Japan, February, 1987

[9] M. Kojima, S. Mizuno, A. Yoshise, "A Polynomial-Time Algorithm For a Class of Linear Complementarity Problems", *Report No. B-193, Department of Information Sciences, Tokyo Institute of Technology*, Tokyo, Japan, June, 1987

[10] M.K.Kozlov, S.P. Tarasov, L.G. Khachiyan, "Polynomial Solvability of Convex Quadratic Programming", *Doklady Akademiia Nauk SSSR 248 (translated in Soviet*

Mathematics Doklady 20, 1108-1111, 1979)

[11] N. Meggido, "Pathways to the Optimal Set in Linear Programming", *Research Report, IBM Almaden Research Center*, San Jose, CA, 1986

[12] R. C. Monteiro, I. Adler, "An $O(n^3L)$ Primal-Dual Interior Point Algorithm For Linear Programming", *Report ORC 87-4, Operations Research Center, Department of Operations Research, University of California*, Berkeley, CA, May 1987

[13] J. Renegar, "A Polynomial-Time Algorithm Based on Newton's Method for Linear Programming", *Report MSRI 07 118-86, Mathematical Sciences Research Institute*, Berkeley, California, June, 1986

[14] P.M. Vaidya, "An Algorithm for Linear Programming Which Requires $O(((m + n)n^2 + (m + n)^{1.5}n)L)$ Arithmetic Operations", *preprint, AT&T Bell Laboratories*, Murray Hill, N.J., 1987

[15] Y. Ye, E. Tse, "A Polynomial Algorithm For Convex Quadratic Programming", *manuscript, Department of Engineering-Economic Systems, Stanford University*, Stanford, CA, 94305, 1986

END

DATE

FILMED

DEC.

1987